# Relevance of Similarity Measures Usage for Paraphrase Detection

Tedo Vrbanec[1,2][a] and Ana Meštrović[2][b]
[1]*Faculty of Teacher Education, University of Zagreb, Savska Cesta 77, 10000 Zagreb, Croatia*
[2]*Department of Informatics, University of Rijeka, Radmile Matejčić 2, 51000 Rijeka, Croatia*
*tedo.vrbanec@ufzg.unizg.hr, amestrovic@inf.uniri.hr*

Keywords: Plagiarism, Deep Learning, Natural Language Processing, Text Similarity, Distance Measures.

Abstract: The article describes the experiments and their results using two Deep Learning (DL) models and four measures of similarity/distance, determining the similarity of documents from the three publicly available corpora of paraphrased documents. As DL models, Word2Vec was used in two variants and FastText in one. The article explains the existence of a multitude of hyperparameters and defines their values, selection of effective ways of text processing, the use of some non-standard parameters in Natural Language Processing (NLP), the characteristics of the corpora used, the results of the pairs (DL model, similarity measure) processing corpora, and tries to determine combinations of conditions under which use of exactly certain pairs yields the best results (presented in the article), measured by standard evaluation measures Accuracy, Precision, Recall and primarily F-measure.

## 1 INTRODUCTION

Plagiarism is a cancer of the academic community, and the fight against it is constantly enhanced, both quantitatively and qualitatively, influenced by three factors: (a) the trend for open access to scientific and professional publications, (b) the requirements of the academic community that any publication must be checked for potential plagiarism and (c) constantly improving the plagiarism detection software. News about the discovery of the plagiarism case is published almost every day. Plagiarism is illegal in most countries and is sanctioned in ordered societies. Plagiarism may arise and manifest in a multitude of ways and shapes, and one of them is a non-referenced paraphrase. Paraphrasing is a form of plagiarism that is often difficult to detect because it changes the structure of the sentence, words with their synonyms, expressions with those of similar or the same meaning, etc. Plagiarism detection software is mostly unable to recognize paraphrasing because they seek similarities mainly through texts alignment. So, for the further progress of detecting the similarity of the texts, it is necessary to extract their semantics and compare them at that level. The focus of the presented long-term research and its results is in the use of DL language models, ie the use of the results of unsupervised learning of neural networks, based on their training with the texts of the annotated paraphrased documents, creating vector spaces (VS) of words, ie words and documents embeddings, which can then be used in conjunction with similarity and distance measures to determine the semantic similarities of lexically different texts.

Research on ways, approaches and models of determining paraphrased texts or their parts, is searching for answers to a multitude of questions, such as: how different measures of similarity or distance affect the results resulting from the use of different DL models or are some combinations of pairs (measure, model) more effective than others and in which contexts? The article gives the answer to these questions and a multitude of other related issues.

After this introduction, the second section presents rare articles whose theme is the interdependence of measures and models. In the third section, the Materials and Methods are presented with the experiments implemented. In the fourth section, the methodology of preparation, processing of results and their evaluation, referring to used corps, text processing and hyperparameters and used DL models

[a] https://orcid.org/0000-0002-4934-255X
[b] https://orcid.org/0000-0001-9513-9467

Vrbanec, T. and Meštrović, A.
Relevance of Similarity Measures Usage for Paraphrase Detection.
In *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2021) - Volume 1: KDIR*, pages 129-138
ISBN: 978-989-758-533-3; ISSN: 2184-3228

are presented. In the fifth section, the general measures of similarity and distance used in the paper are briefly presented with equations and graphic illustrations. In the sixth section, all results of the research and experiments are presented, the seventh section gives conclusions that can be brought out from the presented results. In the eighth section, implications for further studies and discussion of some topics are described.

## 2 RELATED WORK

Harispe et. al. first, then Luu et. al. gave an extensive overview of the text distance and similarity measures intending to find the similarity of the websites, but without the use of the vector presentation of the text (Harispe et al., 2015, 2016; Luu et al., 2020), except for Cosine Similarity measure where this is necessary. They concluded that Cosine Similarity is not the best choice in some cases, because it is less effective. Harispe et al. also noted that it is difficult to estimate the semantic similarity between two words using corpus-based measures.

Sidorov et al. proposed a similarity measure they called Soft Cosine Similarity (Sidorov et al., 2014), which in addition to basic Cosine Similarity considers the similarity between features, features that are known and do not need to be learned from the data. When there is no similarity between features then Soft Similarity Measure is equal to the standard Cosine Similarity, as presented in Equation 6, Section 5.

Charlet & Damnati used the DL Word2Vec model in combination with Soft Cosine Similarity to get semantic similarity of texts (Charlet & Damnati, 2017). They explored unsupervised similarity measures, using the Word2Vec model trained on English Wikipedia, with 300 dimensions of Vector Space (VS). They tried to vary the number of dimensions, but that did not provide them with any significant difference.

Mohammad & Hirst tried to get some semantic similarity measures, so they focused on using ontology-based measures and distributional measures for it (Mohammad & Hirst, 2012a, 2012b). They faced significant research problems doing it.

In their struggle with plagiarism and paraphrasing detection, seeking to extract corpus-based semantic similarities from texts, Vrbanec & Mestrovic tried to identify means, tools and measures for it and then they performed means of experiments in which they calculated similarities between texts using several corpus-based DL models paired with Cosine Similarity measure only (Vrbanec & Meštrović, 2017, 2020, 2021).

Many studies are presenting usage of DL models as well as those using measures of distance and similarity, but to the best of our knowledge, none of them dealt with their pairing, seeking a possible relationship in terms of direction and intensity of relationships.

## 3 MATERIALS AND METHODS

The questions from the introduction can be answered by collecting data obtained from a multitude of experiments. For their implementation, hardware, software and public resources such as an annotated corpus of paraphrased documents are required.

The experiments were performed on computers with the Debian OS ver. 9-11. Programming language Python (ver. 2.7-3.9) was used to design the experiments. Python together with C ++ is considered a de facto standard for NLP, whereby Python prevails in the number and trend of implementations. At the beginning of the research, many necessary libraries have not yet been developed for version 3. Therefore the program development went with ver. 2.7. In time, the required libraries were developed for Python 3, and they stopped developing and became obsolete or unavailable for Python 2, so the program development continued in version 3. Many open-source libraries have been used, and the most important among them are *numpy, gensim, matplotlib, nltk, sklearn, pandas, tensorflow, transformers* and *cython*. All of them are optimized by their authors and in the background, they use the C ++ code, so the Python program was working at a satisfactory speed. Where needed, multiprocessing was used.

Three computers were used as the hardware base. Two typical laptops with dual-core i7 7th generation and quad-core i5 10th generation CPU's were primarily used, with 20GB and 12GB of RAM, without additional support for parallel processing (with integrated Intel HD Graphics). Relatively modest hardware resulted in the need for maximum optimization of the program code. One smaller part of the job was performed on a 48-core workstation, with 2TB RAM and 3 mighty Nvidia Quadro RTX 6000 24GB of GDDR6 graphics cards. Despite this respectable power, for pairs (measures, model) that have already been known as process demanding, it took a lot of time for their processing, so it was ultimately confirmed that some measures (Word Mover's distance and Levenshtein distance) or DL models (Embeddings from Language Models - Elmo) are overwhelming for today's common computer power, they are therefore practically unusable and should be dismissed from pragmatic reasons. The same applies to (Soft Cosine Similarity, Glove Words) pair (measure, model). In addition, the results of these (measures, model) of the pairs that the workstation calculated after several weeks are not top-level so that their rejection is not a big loss.

# 4 METHODOLOGY

The experiments were used to obtain the similarity of the documents from the three publicly available corpora of text documents, which contained paraphrased documents and had annotations about them (ground truth). In real life, it is usually necessary to examine the similarity of a document with all remains from a set of documents, so it was the logic of experiments that were conducted. Below we will show corpora, NLP processing, hyperparameters and DL models.

## 4.1 Corpora

Four corpora were used for experiments. One was made by the authors out of ten different texts that contain a couple of related topics and an intruder text that was dissimilar to the others. This corpus served for the following purposes:

a) Prototyping program.
b) Testing the accuracy of similarity and distance measures calculation.
c) Testing the creation and correctness of the words presented in the form of DL models embedding, created from the texts corpus.
d) Determining the best hyperparameters for experiments.

Since the test corpus was used only to adjust hyperparameters, there was no need to extract the validation subset of datasets for experiments with other corpora. For the smallest (CS) corpus this separation of the data would not be possible, since it contains only 100 texts. So it was necessary to perform cross-validation procedures for the validation of the results (using five pseudo-random samples, each with 20 randomly selected documents from never selected documents for the test dataset, and 80 complementary as train dataset. The remaining two corps are very large, with 10948 (MSRP) and 15718 (Webis11) documents. The usual division on randomly selected 80% - 20% train-test parts was performed.

Some features of the three used corpora are shown in the next Table 1.

Table 1: Some features of three public corpuses.

| Feature | CS | MSRP | Webis |
|---|---:|---:|---:|
| Documents | 100 | 10948 | 15718 |
| Words | 21362 | 210332 | 4928055 |
| Unique Words | 2121 | 17321 | 68658 |
| Max. Words@Docs | 529 | 34 | 4993 |
| Mean Words@Dosc | 213.07 | 19.21 | 320.34 |
| St. Deviation | 77.4 | 5.2 | 272.42 |

From the data shown in Table 1 the exceptional variety of three corpora is visible. This can be both positive and negative because it is possible to evaluate the pairs (model, measure) in different contexts, but also brings the danger that, as we will see later, the results in different contexts are quite different, resulting in difficulties in generalization and conclusion. Box charts in Figure 1, visualised significant parameters of the three corpora.

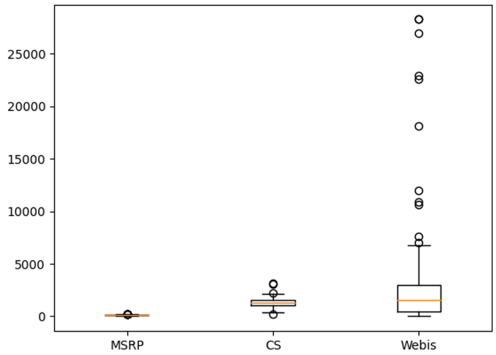

Figure 1: Distribution of word numbers in used corpora.

For the NLP domain, there are several annotated corpora, but their vast majority can not serve to detect paraphrasing. Corpora that were used, CS (Clough & Stevenson, 2009), MSRP (Dolan et al., 2005) and Webis11 (Burrows et al., 2013), were selected and used because they are (a) publicly available and (b) are annotated, ie they contain the official results of human evaluation of paraphrasing. All the other corpora we know that exist are not available, despite contacting the authors who have created and used them. Of those that are available, there are only those that are the supersets of Webis that is big enough alone and it was not reasonable to increase it even more. The exception is Semeval-2014 Task 3 Cross-Level Semantic Similarity Corps, which, with some effort and transformation could serve for the experiments of the semantic similarity detection (Jurgens et al., 2014, p. 2014), although the text units that are compared in it are not of equal sizes.

## 4.2 NLP and Hyperparameters

In the first phase of the experiments, NLP processing was conducted over the documents, but preliminary experiments have shown that standard NLP text processing does not always have to result in the best results.

According to the criterion to contribute to better results (measured by F1), by using preliminary experiments on test corpus, the following hyperparameters or program settings were established:

- Only the single words are used, ie not phrases combined by several words.
- A single word is a minimum number of words in a corpus to take into account when processing text.
- All words are left in processing, even those that have only one letter.
- Stop words were retained in processing, ie they were not rejected.
- The text processing window is 5 words.
- Hypernyms, lemmatization or morphological analysis were not used because their usage did not contribute to the results.
- All numbers in the text are retained.
- 70 iterations were used for neural network training.
- The learning rate of DL models was 0.025.
- In experiments, all types of words were used, ie part-of-speech (POS) tags were not used in experiments.
- To convert the results of distance measures to similarity, the next complementary equation was used:

  $$S = 1 - D \tag{1}$$

  In addition, the angular can be used:

  $$S = 2arccos\frac{D}{\pi} \tag{2}$$

  as well as the base:

  $$S = \frac{1}{1 + D} \tag{3}$$

  but experimental results did not show the need to use the latter two. Regardless of which way of conversion of distance into similarities was used, the results did not change, so the authors could ultimately use the simplest ie complement (1) which is a linear function.
- The mathematical equation implemented in the program, automatically calculates the size of the model, i.e. the number of vectors' dimensions that represent a particular word of a language model as an input parameter of the neural network:

  $$Dim = \min\,(10ceil(log_2 NW), 300) \tag{4}$$

  where the NW is the number of corpus words. The second expression was originally used, which is considerably more complicated and also more difficult to understand, resulting in a smaller number of model dimensions, especially for larger corpora, which was somewhat negatively reflected in the results of some experiments.
- Due to the existence of outliers, documents in corpora whose number of words is much larger than the arithmetic mean (AS), a sigma deviation has been introduced from the number of words in a document/text. The value of two sigmas was initially used, but since the results did not differ from those without sigma limitations, the value of sigma was reduced to 0.5, without statistical relevant negative effects on the results.
- Sigma activator was introduced, a parameter that activates the sigma reduction only if the document has a minimum number. It was originally defined by 10000 words, then 5400 words, finally 3000 words, and according to the results, it would probably be less, but remained so from logical precaution, not because of the needs of better results. The explanation for 3000 characters: 1800 characters are one text card and it is equivalent to 250-300 English words, ie 3000 words should match 10 standard pages of clean text.
- For normalized and non-normalized vectors, cosine similarity has the same value. But any vector operations may negatively affect the final result if vectors are previously normalized (Mohr, 2021). However, the results of the experiments with normalization or without normalization of documents show that normalization has a positive effect on results. Similar, for some DL models, it is necessary to form the documents vectors from the vectors of the words. In that case, the vectors mean or sum of vectors can be used. Although the sum is logically a better option because different amplitude vectors may have significance for their distinction, the results of the experiments showed the opposite.

For each pair of documents, the results of similarity were calculated. The results were compared with official, i.e. calculated Accuracy, Precision, Recall and F1 measure. The results were sorted according to decreasing F1 measures, for each measure of similarity.

### 4.3 DL Models Used

DL models have proved to be very effective for solving problems in computer analysis of natural languages because they create very high-quality vector representations of words, which, in addition to syntactic similarity can be measured semantic. Since DL models create a vector space of words, sentences or phrases with embedded semantic significance, they have the potential for use in finding semantic similarities between documents, e.g. those that have been paraphrased. Word2vec was used in the conducted experiments in its two variants Skip Gram (SG) and Continuous Bag of Words Model (CBOW), and FastText as ways to create embeddings words.

The **Word2Vec** model (Mikolov et al., 2013) uses machine learning through a neural network and with unattended learning from the words creates a vector space of the words embeddings. For it, several parameters have been used, of which the most important ones are: a dimension of VS, the minimum frequency of words, window size and learning speed. Word2Vec consists of two submodels. CBOW (Continuous Bag-of-Word Model) predicts a missing word if the context of missing words is presented to the model. The skip-gram submodel predicts the context of the presented word and reveals an analogy between words.

Bojanowski et al. have noticed that the vector spaces of the words ignore the morphology of the words. Therefore, different words are given different vectors, which is a limitation, especially for flexible and morphologically rich languages that are characterized by large dictionaries and a large number of rare words (Bojanowski et al., 2016). Their **FastText** model just like Word2Vec uses a continuous input of the words and trains the model on a large-scale corpus, but with the difference that each word is presented as a bag of n-grams of characters. The vector representation is connected to each n-gram of characters, so the words are represented by the sum of the vector representation of the n-gram of characters.

To use DL models, the n-dimensional vector for each document must be extracted from the model or if a model has no document vector, it must be made from the individual word vectors. Some DL models can obtain both word vectors and documents vectors, while others have only one of those abilities. Both used models do not provide the ability to obtain document vectors, so their words vectors were used to make a combination that would present the entire documents. This has been done in two possible ways, by the sum or the average value of the word vectors. So far there is still no better-documented way of creating a document vector from the words vectors.

## 5 SIMILARITY AND DISTANCE MEASURES

Several common measures of similarity and distance have been used in experiments. We will briefly present them in this section. For two n-dimensional vectors in the Vector Space Model (VSM), A=($A_1$, $A_2$, … $A_n$) and B=($B_1$, $B_2$, …, $B_n$), which represent a document, we used standard similarity measures:

**Cosine Similarity** is obtained from the dot product of vectors and is defined as the cosine function of the angle between two non-null-vectors.

$$cos\varphi = \frac{\sum_1^n A_i B_i}{\sqrt{\sum_1^n A_i^2}\sqrt{\sum_1^n B_i^2}} \tag{5}$$

**Soft Cosine Similarity** measure in addition to the vector components takes into account the similarities between the pairs of features**.** (Sidorov et al., 2014).

$$softcos(A,B) = \frac{\sum_{i,j}^n s_{i,j} A_i B_j}{\sqrt{\sum_{i,j}^n s_{i,j} A_i A_j}\sqrt{\sum_{i,j}^n s_{i,j} B_i B_j}} \tag{6}$$

Where $s_{ij} = sim(f_i, f_j) = cosine(e^i, e^j)$ represent similarities of feautres, and $e^k = (0, \ldots, 1, \ldots, 0)$ are the elementary vectors representing words. The relationship between these two measures is illustrated in Figure 2 (Novotný, 2021).

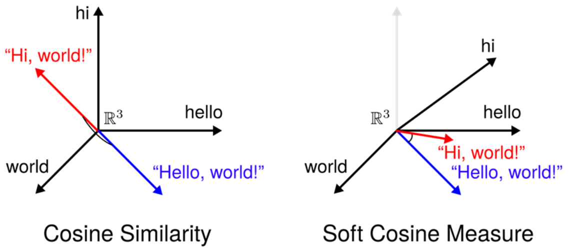

Figure 2: Cosine measures (Source: tinyurl.com/23dvvvps).

**Euclidean Distance** is the shortest distance between two points in Cartesian Space, therefore can be calculated by expanding the Pythagorean theorem to n-dimensional VS. Euclidean distance and its difference from the Cosine Similarity is shown in Figure 3.

$$euclidean(A,B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \tag{7}$$

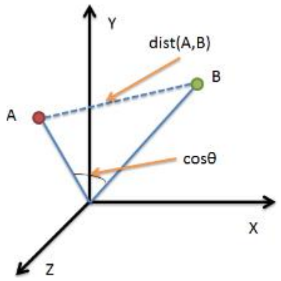

Figure 3: Euclidean Distance and Cosine Similarity (Source: https://images.cnitblog.com/).

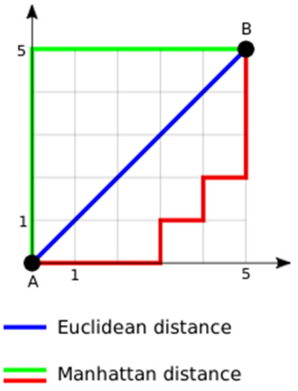

Figure 4: Manhattan and Euclidean Distances (Luu et al., 2020).

**Manhattan Distance** (City Block Distance, Taxicab Distance) is the distance between two points in VSM calculated as the sum of the absolute differences of their Cartesian coordinates. Manhattan distance and its relation to Euclidean Distance is shown in Figure 4

$$manhattan(A,B) = \sum_{i=1}^{n} |A_i - B_i| \tag{8}$$

In addition to the measures shown, Levenshtein distance and Word Mower distance were also used in experiments, but due to excessive demand for computer power, ie practical inability to obtain the results of these distances for the largest corpus, they are not included in the results.

# 6 RESULTS

In addition to determining hyperparameters using a separate corpus, it was necessary to solve the problem of outliers by reducing their size (number of words). It is necessary to balance two contradictory requirements: (a) that the smaller reduced size maximizes computer processing and (b) that their reduced size minimally affects the results. This problem was solved by introducing two parameters: sigma and sigma activator.

The *sigma* parameter defines that the size of the outliers decreases to the specific number of words that is limited by the arithmetic mean + *sigma* values. Pre-research experiments have shown that there is no effect on the results if the value of the *sigma* parameter reduces to 0.5. Further reduction was not implemented because authors wanted to avoid a significant impact on the corpora. All experiments were conducted with this value. For documents with a small number of words, there is no need for the reduction of their number of words. Therefore, the *sigma activator* parameter has been introduced activating reducing the number of words only for those documents that have a minimum number of words.

With this research, the authors wanted to determine to what extent the number of dimensions of DL embeddings affects the results, as other researchers used various dimension numbers, without arguing why they use a particular number and usually defined it as a fixed size. For example, the Elmo DL predefined model whose embeddings can be downloaded from the Internet uses 512 and 1024 dimensions (Peters et al., 2018), Google has a multitude of models like LangBERT (Feng et al., 2020) using 768 dimensions. For these reasons, six series of experiments for all three used corpora were carried out, with an increased number of dimensions, and the reduced sigma activator. The results are shown below.

## 6.1 Model and Corpora Parameters

This section analyzes how the dimensions of DL models and solving the problem of outliers - extremely large files in the corpus, affect the performance of the pairs (measure, model) and compares their success of finding similarities among the documents of the same corpus. Six series of experiments were conducted, which included changes in the two parameters: DL models changed the number of dimensions starting from the minimum (equation calculated) to a maximum of 300, and reducing the sigma activator starting from 10000, over 5400 to 3000 words. 3000 words were determined because one text card contains 1800 characters, which corresponds to 250-300 English words, ie in 3000 words is equivalent to 10 text cards, and this should be enough for the quality training of DL models.

### 6.1.1 CS Corpus Results

The CS corpus consists of only 100 documents, with a maximum of 529 words in them. Therefore, none of the above values of sigma activators has any effects on it. Consequently, for the same number of dimensions, the same model, the same measure of similarity, and different sigma activators, the experiments on this corpus gave the same or very similar results. Differences in the third decimals can be attributed to differences in random initial settings of neural networks during their initialization. The best results for CS corpus arise from 110-dimensional models, and not, to be expected, from the highest dimensions (300), which is explained in the conclusion. The Soft Cosine Similarity measure is completely inert to the number of dimensions and according to the used DL model. For all of them, it gives the same threshold of 0.67 and the same F measure of 73.9%, which is significantly lower than the best combination (Euclidean, Word2Vec CBoW) whose result was 95.4% -95.7%.

In the desire to achieve the best results, in two cases of the largest number of dimensions usage, the models where the documents vectors are made from the word vectors (because the models do not contain document vectors), instead of the normalized average, the non-normalized sum of the word vectors has been used in experiments. This change did not meet the expectations: the results of such calculations are worse than the others.

### 6.1.2 MSRP Corpus Results

The MSRP corpus consists of 10948 documents that have a maximum of 34 and an average of 22 words. Due to the small number of words in documents, as with CS corpus, the sigma activator has no function. Therefore, the same measures with the same models and their dimensions are producing approximately the same results. Robust Soft Cosine Measure that had the average results in the CS corpus, turned into a leader, taking all leading positions in this corpus. It follows it to a small distance of the Word2Vec Skip-Gram model, regardless of the dimensions of the model and distance measures. Other combinations (model, measure) follow them, with a clear trend that increasing the number of dimensions lowers F-measure. It should be noted that the difference between the highest and lowest result was only 2.6%.

### 6.1.3 Webis Corpus Results

Webis-11 (W) corpus is assuredly the least compact and largest of the three that were used. It contains plenty of outliers: out of a total of 17056 documents, 1672 are empty, 15384 remain for processing, with a maximum of 4993 and an average of 320.21 words. Sigma activator with the value of 3000 activates the reduction of 4657 documents that are reduced to average + 0.5 * sigma, ie to the max. 2610 words.

With an important fence that the difference between the highest and lowest value of F-measures is only 1.1% (67.9-66.8%), in this case, the most important factor of performance is the dimensionality of the model: the best results were obtained with the models that had the largest number of dimensions, with evenly participation All similarity/distances measures and with the primacy of Word2Vec Skip-Gram model. Soft Cosine measures had the worst results, with the unusually low threshold of 0.08.

## 6.2 Comparison of Measures

After selecting the best combination (measure, model) in the previous section, in this section, we are dealing with the efficiency of four similarities or distances translated into similarity. In Table 2, the results for the CS corpus are shown.

Table 2: Results of measures using Word2Vec and FastText 110-dimensions models trained on CS corpus.

| Measure | Model | T | A | P | R | F1 |
|---|---|---|---|---|---|---|
| Euclidean | W2V-CB | 0.34 | **0.983** | **0.937** | **0.980** | **0.957** |
| Manhattan | W2V-CB | 0.36 | 0.981 | 0.932 | 0.974 | 0.951 |
| Euclidean | FastText | 0.39 | 0.975 | 0.927 | 0.950 | 0.936 |
| Manhattan | FastText | 0.40 | 0.973 | 0.919 | 0.950 | 0.931 |
| Cosine | W2V-CB | 0.55 | 0.963 | 0.858 | 0.990 | 0.915 |
| Euclidean | W2V-SG | 0.40 | 0.965 | 0.931 | 0.900 | 0.908 |
| Cosine | W2V-SG | 0.82 | 0.965 | 0.931 | 0.900 | 0.908 |
| Cosine | FastText | 0.63 | 0.958 | 0.841 | 0.975 | 0.900 |
| Manhattan | W2V-SG | 0.39 | 0.959 | 0.934 | 0.840 | 0.884 |
| Soft Cosine | FastText | 0.67 | 0.907 | 0.832 | 0.683 | 0.739 |
| Soft Cosine | W2V-CB | 0.67 | 0.907 | 0.832 | 0.683 | 0.739 |
| Soft Cosine | W2V-SG | 0.67 | 0.907 | 0.832 | 0.683 | 0.739 |

Legend:
W2V-CB = Word2Vec CBoW, W2V-SG = Word2Vec Skip-Gram
T, A, P, R = Threshold, Accuracy, Precision, Recall

We can see that under the same conditions various measures of similarity/distance have quite different results, preferring Word2Vec CBow as a model and Euclidean and Manhattan measures in front of the (Soft) Cosine. The difference between the best results and the worst is significant: 95.7% -73.9% = 21.8%. The CS corpus is the most similar to what we would expect from a production system by the number of documents and their size. Such a system should have a module that finds thematically similar documents and compares their similarity to the checked document.

Table 3 shows the results for the MSRP corpus. With its form of a lot of small texts up to 40 words, it may be suitable for finding smaller similar passages or sentences from filtered documents previously obtained in some way. We see that the soft cosine measure has the best results, and it is very interesting to notice again that it is inert to the model used. The results vary in the interval of only 1.1% (81.6% - 79.5%).

Table 3: Results of measures using Word2Vec and FastText 88-dimensions models trained on MSRP corpus.

| Measure | Model | T | A | P | R | F1 |
|---|---|---|---|---|---|---|
| Soft Cosine | W2V-SG | 0.48 | **0.714** | **0.705** | **0.967** | **0.816** |
| Soft Cosine | W2V-CB | 0.48 | **0.714** | **0.705** | **0.967** | **0.816** |
| Soft Cosine | FastText | 0.48 | **0.714** | **0.705** | **0.967** | **0.816** |
| Euclidean | W2V-SG | 0.64 | 0.702 | 0.701 | 0.951 | 0.807 |
| Cosine | W2V-SG | 0.88 | 0.693 | 0.692 | 0.957 | 0.804 |
| Manhattan | W2V-SG | 0.63 | 0.692 | 0.690 | 0.963 | 0.804 |
| Euclidean | FastText | 0.61 | 0.690 | 0.696 | 0.936 | 0.799 |
| Cosine | FastText | 0.83 | 0.684 | 0.685 | 0.960 | 0.799 |
| Manhattan | FastText | 0.59 | 0.678 | 0.678 | 0.968 | 0.798 |
| Cosine | W2V-CB | 0.81 | 0.668 | 0.666 | 0.990 | 0.796 |
| Euclidean | W2V-CB | 0.56 | 0.667 | 0.665 | 0.991 | 0.796 |
| Manhattan | W2V-CB | 0.60 | 0.669 | 0.670 | 0.978 | 0.795 |

Legend:
W2V-CB = Word2Vec CBoW, W2V-SG = Word2Vec Skip-Gram
T, A, P, R = Threshold, Accuracy, Precision, Recall

Table 4 shows results for measures and models for Webis corpus that is very diverse and has large deviations. Since the results vary only 0.5% (67.9-67.4%) it is difficult to make conclusions. We note that the models are evenly distributed to the ranking list and that Manhattan and Euclidean measures have better results than the (Soft) Cosine measures, which is following the results of the CS corpus, at least in the relative ranking relationships.

Table 4: Results of measures using Word2Vec and FastText 300-dimensions models trained on Webis corpuses.

| Measure | Model | T | A | P | R | F1 |
|---|---|---|---|---|---|---|
| Manhattan | W2V-SG | 0.95 | **0.540** | **0.532** | 0.939 | **0.679** |
| Euclidean | W2V-SG | 0.95 | **0.540** | **0.532** | 0.939 | **0.679** |
| Euclidean | FastText | 0.93 | 0.538 | 0.531 | 0.938 | 0.678 |
| Manhattan | W2V-CB | 0.92 | 0.537 | 0.530 | **0.940** | 0.678 |
| Euclidean | W2V-CB | 0.92 | 0.537 | 0.530 | **0.940** | 0.678 |
| Manhattan | FastText | 0.92 | 0.535 | 0.529 | 0.943 | 0.678 |
| Cosine | W2V-CB | 0.40 | 0.524 | 0.522 | 0.962 | 0.677 |
| Cosine | W2V-SG | 0.40 | 0.524 | 0.522 | 0.962 | 0.677 |
| Cosine | FastText | 0.11 | 0.523 | 0.522 | 0.963 | 0.677 |
| Soft Cosine | FastText | 0.08 | 0.522 | 0.521 | 0.953 | 0.674 |
| Soft Cosine | W2V-CB | 0.08 | 0.522 | 0.521 | 0.953 | 0.674 |
| Soft Cosine | W2V-SG | 0.08 | 0.522 | 0.521 | 0.953 | 0.674 |

Legend:
W2V-CB = Word2Vec CBoW, W2V-SG = Word2Vec Skip-Gram
T, A, P, R = Threshold, Accuracy, Precision, Recall

# 7 CONCLUSION

Questions and conclusions related to this research can be grouped along the models, measures and vector forms representing texts.

Redundant **dimensionality** harms the results. Although we have considered that it is necessary to have as much (though not too high) the dimensionality of embeddings of the DL model (300-dimensional vectors) to obtain the best results, it turned out to be worth using the algorithm that calculates the number of dimensions for corpora (depending on the number of words in the corpus), because approximately the optimal number of dimensions produced good, even better results with less computer processing and obtaining fewer data.

The results show that the model dimension is the most important parameter for obtaining the best results, where the dimension should not be determined according to the principle of one amount for all corpora but should be dynamically counted for each corpus. Therefore, a mathematical function was proposed, which was obtained based on empirical results. Since the small CS corpus that had only 100 documents of the average size of a text card, corresponding to the modification of the equation for the calculation of the dimensionality, which increased the dimensionality of 72 to 110, and since the other (MSRP corpus) that has a large number of very short texts, corresponding to the same number of dimensions, and since the largest corpus with a large number of texts among which are very large ones with over 20 text cards, corresponding to the maximum number of dimensions, we conclude that the key parameter of the couple's success (measure, model) is the equation for calculating the dimensions of DL models based on the number of words.

Increasing the number of DL models above a certain value, which depends on the size of the corpus or the number of words, has a negative correlation with the results. It is also interesting that although the amount of data for those multidimensional models that had an unnecessarily large number of dimensions increases, their compressed record does not grow (data has been serialized using the Python module *cpickle*, then compressed to the *gzip* format), which is an additional proof that such records contain redundancies that compression algorithms recognize.

Based on the obtained and presented results, in the offer of **measures** and models presented in this paper, we would surely give an advantage of Euclidean and Manhattan measures according to the (Soft) Cosine measures and Word2Vec models compared to FastText, with a fence that if we want to further analyze the relationship between the similarity of the short text units, Soft Cosine measure can not be discarded and has great potential. Soft Cosine measure is a model-resistant, ie, regardless of the input DL model and its dimensions, it provides equal results (at least for CS and MSRP corpora).

In models that create embeddings for words, but not for whole texts, **vector presentations** of texts are necessary to obtain as a linear combination of words vectors. The average value proved to be better than the sum of the word vector. Similarly, the normalization of the resulting documents vectors gives better results.

## 8 DISCUSSION

For further studies, a well-annotated corpus for paraphrasing is required. The authors of this paper will present the corpus and make it publicly available. That corpus will contain 100 documents and their paraphrases. The existing two major MSRP and Webis corpora are partially mis-annotated, because (part of) the annotation is done through Amazon Mechanical Turk which is not well done, and that can be easily proved by insight into official results (human evaluations) and their comparisons with pairs of texts that are evaluated. Furthermore, the authors plan to adapt Semeval-2014 Task 3 Cross-level Semantic Similarity Corpus and carry out experiments over it, as well as on the P4PIN corpus pointed by the reviewer.

The paper proposed one possible mathematical equation for calculating the near-optimal number of dimensions of DL models. But can it be better? With new corpora, it is possible to further check and then modify it. It is certain that the dimension of the model depends on the size of the input corpus on which the models are trained. We will repeat experiments using the equation

$$Dim = log_2 NUW \cdot log_2 ND \tag{9}$$

where is NUW number of unique words, and ND is the number of documents in the corpus. The logic for such a concept of the equation is as follows: (a) All unique words are required to encode into unique binary records (the first part of the equation) and (b) each word can be located in many contexts, depending on the size of the corpus (the second part of the equation).

Future experiments could be repeated with a larger observation window when training DL models, as well as part-of-speech tags could be used, where the most promising is the usage of nouns and verbs.

Experiments should also be carried out with other DL models, such as Doc2Vec, GloVe, USE, ELMo and BERT. Only two DL models have been used in the article because too much data would make it difficult to create conclusions, given that the emphasis of the article was an impact on the results of various similarity/distances measures.

## ACKNOWLEDGEMENTS

This research was funded by the University of Rijeka grant number uniri-drustv-18-38.

## REFERENCES

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching Word Vectors with Subword Information. *Computing Research Repository*. https://doi.org/1511.09249v1

Burrows, S., Potthast, M., & Stein, B. (2013). Paraphrase acquisition via crowdsourcing and machine learning. *ACM Transactions on Intelligent Systems and Technology*, *4*(3), 1. https://doi.org/10/gbdd2k

Charlet, D., & Damnati, G. (2017). SimBow at SemEval-2017 Task 3: Soft-Cosine Semantic Similarity between Questions for Community Question Answering. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 315–319. https://doi.org/10/gjvjk5

Clough, P., & Stevenson, M. (2009). Creating a Corpus of Plagiarised Academic Texts. *Proceedings of the Corpus Linguistics Conference*, *January 2009*.

Dolan, B., Brockett, C., & Quirk, C. (2005). *Microsoft Research Paraphrase Corpus*. Microsoft Research. https://www.microsoft.com/en-ca/download/details.aspx?id=52398

Feng, F., Yang, Y., Cer, D., Arivazhagan, N., & Wang, W. (2020). *Language-agnostic BERT Sentence Embedding*. http://arxiv.org/abs/2007.01852

Harispe, S., Ranwez, S., Janaqi, S., & Montmain, J. (2015). Semantic Similarity from Natural Language and Ontology Analysis. *Synthesis Lectures on Human Language Technologies*, *8*(1), 1–254. https://doi.org/10/gc3jtd

Harispe, S., Ranwez, S., Janaqi, S., & Montmain, J. (2016). Semantic Measures for the Comparison of Units of Language, Concepts or Instances from Text and Knowledge Base Analysis. *ArXiv:1310.1285 [Cs]*. http://arxiv.org/abs/1310.1285

Jurgens, D., Pilehvar, M. T., & Navigli, R. (2014). SemEval-2014 Task 3: Cross-Level Semantic Similarity. *SemEval@ COLING*, 17–26.

Luu, V.-T., Forestier, G., Weber, J., Bourgeois, P., Djelil, F., & Muller, P.-A. (2020). A review of alignment based similarity measures for web usage mining.

*Artificial Intelligence Review*, *53*. https://doi.org/10/gjvnj5

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *ArXiv Preprint ArXiv:1301.3781*. https://arxiv.org/abs/1301.3781

Mohammad, S. M., & Hirst, G. (2012a). Distributional measures as proxies for semantic relatedness. *ArXiv Preprint ArXiv:1203.1889*.

Mohammad, S. M., & Hirst, G. (2012b). Distributional measures of semantic distance: A survey. *ArXiv Preprint ArXiv:1203.1858*.

Mohr, G. (2021). *Vector normalisation: Yes/No or When Y/N?* [Https://groups.google.com/g/gensim]. Gensim. https://groups.google.com/g/gensim/c/-RcUZDp_kq4/m/piaJCL4dAgAJ

Novotný, V. (2021). *Soft Cosine Tutorial*. GitHub. https://github.com/RaRe-Technologies/gensim

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. https://doi.org/10/gft5gf

Sidorov, G., Gelbukh, A., Gómez-Adorno, H., & Pinto, D. (2014). Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. *Computación y Sistemas*, *18*(3), 491–504. https://doi.org/10/gcpzhs

Vrbanec, T., & Meštrović, A. (2017). The struggle with academic plagiarism: Approaches based on semantic similarity. *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2017 - Proceedings*. https://doi.org/10/gj26vx

Vrbanec, T., & Meštrović, A. (2020). Corpus-Based Paraphrase Detection Experiments and Review. *Information*, *11*(5), 241. https://doi.org/10/ghjtff

Vrbanec, T., & Meštrović, A. (2021). Taxonomy of academic plagiarism methods. *Journal of the Polytechnic of Rijeka*, *9*(1), 283–300. https://doi.org/10.31784/zvr.9.1.17.